AD 295 952

DETAB-X AND THE WORLD OF BANKING

Solomon L. Pollack

February 1963

P-2697

DETAB-X AND THE WORLD OF BANKING

Solomon L. Pollack*

The RAND Corporation, Santa Monica, California

Jim Hammond's talk has described what decision tables are. I would like to tell you about a specific decision-table language, DETAB-X (Decision Tables, Experimental), an experimental language that combines COBOL-61 and decision-tables. The CODASYL Systems Group has designated DETAB-X as an experimental language in order to emphasize that it is available on a test basis to those in the business data processing or scientific field who are willing to experiment with it. Hopefully, users of the language will provide feedback concerning its merits and defects to the CODASYL Systems Group.

As a framework for describing DETAB-X, let us consider the three major steps (shown in Slide 1) that result in a computer program. First, system analysts develop the source program, which describes the data in their system, preparatory operations the data must undergo, and the movement of data through the system. Then there is the translator, which converts the source program to a set of computer instructions called an object program; in turn, the object program is processed by the computer to update files and produce required reports.

Slide 2 gives a more detailed breakdown of these elements. Here we see that the source program consists of two major parts: the

---

*Any views expressed in this paper are those of the author. They should not be interpreted as reflecting the views of The RAND Corporation or the official opinion or policy of any of its governmental or private research sponsors. Papers are reproduced by The RAND Corporation as a courtesy to members of its staff.

This paper was prepared for presentation at the 15th Annual Conference of the National Association of Mutual Savings Banks, February 14, 1963, at Boston, Massachusetts.

STAGES IN THE DEVELOPMENT OF A COMPUTER PROGRAM

- BUSINESS-ORIENTED PROBLEM DESCRIPTION
  - DATA DESCRIPTION
  - DATA TRANSFORMATIONS
- PROCEDURES DESCRIPTION
  - EQUIPMENT CONFIGURATION
  - ADDITIONAL DATA TRANSFORMATIONS
- TRANSLATION TO COMPUTER PROGRAM
  - COMPILER
  - PROGRAMMER

Slide 1

FEATURES OF A PROBLEM-ANALYSIS TECHNIQUE

- ORDERLY METHOD OF DOCUMENTING
- INDEPENDENT OF PROCESSING METHOD
- FLEXIBILITY FOR CHANGING PORTIONS OF THE ANALYSIS
- ABILITY TO PORTRAY COMPLEX RELATIONSHIPS AND ALTERNATIVES
- EASE OF REVIEW FOR OMISSIONS AND INCONSISTENCIES

Slide 2

business-oriented problem description and the procedures description. The former describes what needs to be done in the business system; the latter describes how the data processing is to be accomplished. At present, all languages, including DETAB-X, intermix the what and how of data processing. It is the System Group's hope that DETAB-X will lead to a language that will enable the system designer to describe his application separately from his procedures. Then, if a change occurs, the system designer can implement it by changing the problem description, without having to rewrite the entire computer source program.

For the remainder of this talk, we will focus our discussion of DETAB-X on its use for business-oriented problem description. As a prelude, it will be useful to bear in mind the desirable features for a problem-analysis technique (see Slide 3).

Any such technique must first provide an orderly method of documentation. People who have been responsible for computer programs know how devastating it can be when a programmer who has done little or no documentation of his programs leaves the organization. A good technique will make it relatively simple to document a job. DETAB-X can serve this role in business data processing.

A problem-analysis technique that is independent of the processing method enables the statement of the problem to retain its usefulness even if the processing method changes later on. Too often, in the past, the entire application had to be restated when one computer replaced another, only because the processing method was interwoven with the statements of what needed to be done.

SAMPLE NARRATIVE FORM OF DEPOSIT TRANSACTIONS

IF TRANSACTION IS QUERY THEN IF NUMBER IS NOT VALID DISPLAY REJECT-MESSAGE; OTHERWISE IF TRANSACTION-CODE EQUALS DEPOSIT GO TO DEPOSIT-SECTION.

IF TRANSACTION-CODE EQUALS DRAFT GO TO DRAFT-SECTION.

IF TRANSACTION-CODE EQUALS CASH-CHECK GO TO CASH-CHECK-SECTION.

IF TRANSACTION-CODE EQUALS UPDATE-BOOK GO TO UPDATE-BOOK-SECTION; OTHERWISE GO TO NON-QUERY-SECTION.

Slide 3

When an analysis technique provides the system designer with the flexibility to change portions of his analysis, he has greater freedom of action in documenting the application. For example, if the analysis technique does not imply that actions must be performed in the sequence in which they are written, the designer is free to move from application to application, documenting as he goes along. He can then later determine the order of processing based on parameters he selects.

In any banking system, sets of actions occur as a result of a combination of many conditions; consequently, it is highly desirable to have formats that help the analyst visualize the resultant complex relationships. As we shall show later, the decision-table format can prov extremely useful in this regard.

The failure to consider one or more conditions is a continual threat to those responsible for developing data processing systems. An omission in the problem statement may not be discovered until the system has been operating many months. Consider a savings-account system being processed on a computer. A series of actions have been specified for each type of transaction designated by the system designers; but -- people being what they are -- it is possible that one kind of transaction was not provided for. The bank manager would certainly want the computer system to reject the first one of those transactions to enter the system, so that he could take steps to process them properly. Decision-table techniques make it relatively easy to review any system description for omissions and inconsistencies.

DETAB-X prescribes COBOL-61 as the language to be used in a decision-table structure. This is to enable those who intend to process

their applications on computers to take advantage of the many COBOL-61 compilers (translators) that are currently becoming available from the computer manufacturers; however, DETAB-X can also be useful for banking systems that are not tied in with computers.

To illustrate the features of DETAB-X, we first describe a banking function in the conventional form (see Slide 4) and then describe that same function in decision-table form (see Slide 5).

Note that the narrative form (see Slide 4) is serial in nature. The comparisons and the actions based on those comparisons must occur in the order in which they are specified. Note also that this form does not lend itself easily to analysis or to checks for completeness and accuracy. It is difficult to tell whether all the appropriate comparisons on transaction-type, number, and transaction-code have been made. Also, if a comparison is made against several values, it is very difficult to spot wrong values, because corresponding values appear in different paragraphs, some distance from each other.

Let us turn to Slide 5 which shows these same rules in decision-table form. Notice that when the conditions are laid out in tabular form, the system designer is better able to determine if he has considered all the possible combinations of conditions that might occur. He knows, for example, that if there are two conditions that can be satisfied or not satisfied (lines 1 and 2 of Slide 5) and one condition that can assume any one of four values (line 3 of Slide 5) then a total of $2^2 \times 4 = 16$ rules can be formed.

Rule 1 is equivalent to 4 rules by virtue of the dash on line 3. Hence, Rules 1-5 represent a total of 8 rules. Since 16 rules are

SAMPLE DETAB-X FORM OF DEPOSIT TRANSACTIONS

	RULE 1	RULE 2	RULE 3	RULE 4	RULE 5	ELSE
TRANS-TYPE EQ QUERY	Y	Y	Y	Y	Y	—
NUMBER IS VALID	N	Y	Y	Y	Y	—
TRANSACTION-CODE EQ	—	DEPOSIT	DRAFT	CASH-CHECK	UPDATE-BOOK	—
DISPLAY REJECT-MESSAGE	X	—	—	—	—	—
GO TO TABLE	—	002	003	004	005	ERROR

Slide 4

GOALS FOR FUTURE BUSINESS LANGUAGES

1. IMPROVED COMMUNICATION AND DOCUMENTATION
2. INCREASED EFFICIENCY OF COMPUTER PROGRAM
3. REDUCED COMPUTER-PROGRAM CHECKOUT TIME
4. INCREASED ACCURACY IN PROBLEM STATEMENT
5. COMPLETENESS OF PROBLEM STATEMENT

Slide 5

possible, 8 rules are missing. In DETAB-X, we represent the missing rules by an ELSE-Decision-Rule. It is the rightmost rule in Slide 5.

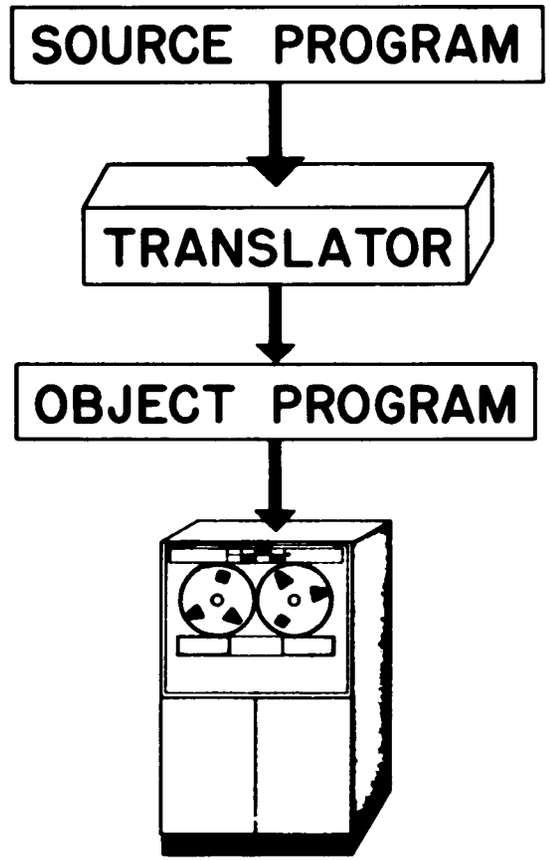
DETAB-X differs from other languages that describe decision rules in that the rules specified in the decision table do not have to be executed in the order they have been written in; that is, rule 1 does not have to be executed first. This gives the compiler freedom to determine the order of rule execution based on some parameter such as frequency of occurrence. For example, if a particular rule is executed 90 percent of the time and the rest only 10 percent, it is certainly more efficient to have that 90 percent rule executed first. The format of DETAB-X makes it easy to specify the parameter for each rule so that more efficient object programs can be developed.

In Slide 6 we have listed some desired goals for future business languages. It is our hope that DETAB-X is a step in this direction. We feel that DETAB-X can help users in documenting their system in that programs written in DETAB-X will provide improved communication between system designers, programmers, and functional specialists. DETAB-X is also expected to increase the accuracy and completeness of problem statement achievable by existing languages. It is available to anyone willing to try it, and the Development Committee would appreciate receiving any information on the merits and defects of the language.*

---

*Criticisms and suggestions concerning DETAB-X should be sent to Solomon Pollack, The RAND Corporation, 1700 Main Street, Santa Monica, California.

DEVELOPMENT OF COMPUTER PROGRAM



Slide 6

BIBLIOGRAPHY

1. Bromberg, Howard, "COBOL and Compatibility," Datamation, February, 1961, pp. 30-34.

2. Department of Defense, COBOL: 1961 Report to CODASYL (Conference on Data System Languages), Government Printing Office, Washington, D.C., 1961.

3. Evans, Orren Y., Advanced Analysis Method for Integrated Electronic Data Processing, IBM General Information Manual #F20-8047, n.d.

4. Grad, Burton, "Tabular Form in Decision Logic," Datamation, July, 1961.

5. Phillips, Charles A., "Current Status of COBOL," Proceedings of USAF World Wiue Data Systems and Statistics Conference, October 26, 1961.

6. Systems Group (CODASYL), Preliminary Specifications of DETAB-X, August, 1962.